Advent of Code [About] [AoC++] [Events] [Settings] [Log Out] icecreambean 22*
y(2016) [Calendar] [Leaderboard] [Stats] [Sponsors]

--- Day 12: Leonardo's Monorail ---

You finally reach the top floor of this building: a garden with a slanted glass ceiling. Looks like there are no more stars to be had.

While sitting on a nearby bench amidst some tiger lilies, you manage to decrypt some of the files you extracted from the servers downstairs.

According to these documents, Easter Bunny HQ isn't just this building - it's a collection of buildings in the nearby area. They're all connected by a local monorail, and there's another building not far from here! Unfortunately, being night, the monorail is currently not operating.

You remotely connect to the monorail control systems and discover that the boot sequence expects a password. The password-checking logic (your puzzle input) is easy to extract, but the code it uses is strange: it's assembunny code designed for the new computer you just assembled. You'll have to execute the code and get the password.

The assembunny code you've extracted operates on four registers (`a`, `b`, `c`, and `d`) that start at `0` and can hold any integer. However, it seems to make use of only a few instructions:

- `cpy x y` copies `x` (either an integer or the value of a register) into register `y`.
- `inc x` increases the value of register `x` by one.
- `dec x` decreases the value of register `x` by one.
- `jnz x y` jumps to an instruction `y` away (positive means forward; negative means backward), but only if `x` is not zero.

The `jnz` instruction moves relative to itself: an offset of `-1` would continue at the previous instruction, while an offset of `2` would skip over the next instruction.

For example:

```
cpy 41 a
inc a
inc a
dec a
jnz a 2
dec a
```

The above code would set register `a` to `41`, increase its value by `2`, decrease its value by `1`, and then skip the last `dec a` (because `a` is not zero, so the `jnz a 2` skips it), leaving register `a` at `42`. When you move past the last instruction, the program halts.

After executing the assembunny code in your puzzle input, what value is left in register `a`?

Your puzzle answer was `317993`.

--- Part Two ---

As you head down the fire escape to the monorail, you notice it didn't start; register `c` needs to be initialized to the position of the ignition key.

If you instead initialize register `c` to be `1`, what value is now left in register `a`?

Your puzzle answer was `9227647`.

Both parts of this puzzle are complete! They provide two gold stars: **

At this point, you should return to your advent calendar and try another puzzle.

If you still want to see it, you can get your puzzle input.

You can also [Share] this puzzle.